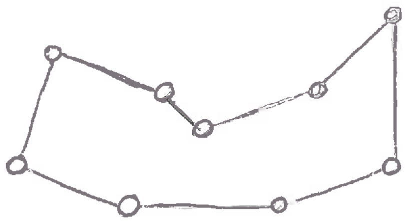
Euclidean Traveling salesman problem (ETSP)

Input : n points in the euclidean plane ($\mathbb{R}^2$).
Task : find the shortest tour visiting all points (order).

e.g. :



<u>Theorem</u>: [Arora '98]
There is a PTAS for TSP in the euclidean plane.

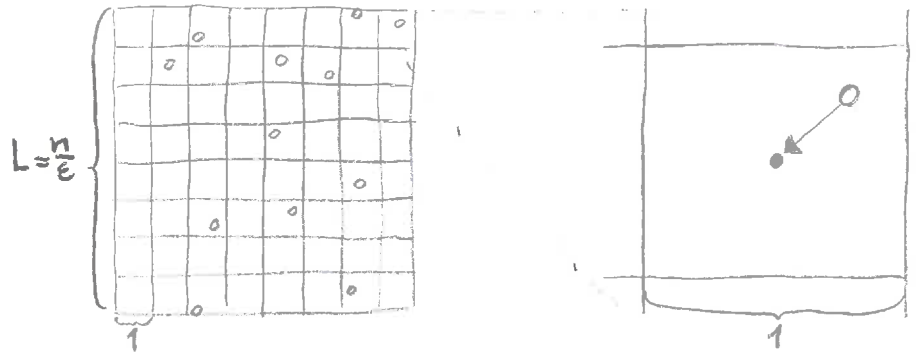
<u>Idea</u>: Show that there is a close to optimal solution that satisfies nice properties, by transforming an optimal solution into one. Then, show that a solution with such properties can be found optimally.

<u>Main steps</u>:
1. Round the instance
2. Restrict the set of solutions
3. Dynamic programming
4. Randomization

1. Rounding (perturbation)

Let B be the smallest square box that is axes-parallel and contains all points. Scale the whole instance such that B has size $\frac{n}{\varepsilon} \times \frac{n}{\varepsilon}$. W.l.o.g., we assume $n/\varepsilon = 2^{k-1}$ with $k = 2\log_2 n/\varepsilon$. Let $L = 2^k$, such that B has size $L \times L$. Introduce a unit grid on B. We move each point of the input to the center of the unit grid square in which it is contained.

Lemma: An optimal order on the rounded instance is a $(1+\varepsilon)$-approximation in the original instance.

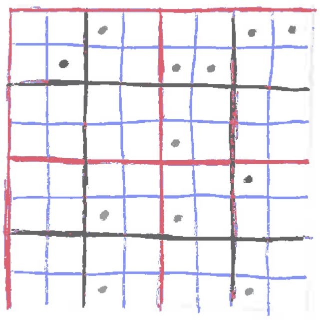


Proof: Exercise

2. Restrict the set of solutions

2a. Quad tree dissection.

We divide the $L \times L$ box repetitively into 4 boxes:



This divides the box into different level squares. B is the level 0 square

4 level 1 squares, size $\frac{L}{2} \times \frac{L}{2}$
16 level 2 squares, size $\frac{L}{4} \times \frac{L}{4}$
$\vdots$
4^i level i squares, size $\frac{L}{2^i} \times \frac{L}{2^i}$
L^2 level k squares, size 1×1

Level i lines: lines introduced in iteration i.

Idea: squares are cells of the dynamic programming table.

2b. Portals

Let $m = \lceil \frac{k}{\varepsilon} \rceil$. For each i, place m equidistant portals on sides of a level i square that are part of a level i line.

A tour is portal respecting if it crosses lines only at portals.

Lemma: An optimal portal respecting tour crosses each portal at most twice.

Proof: Suppose not, the $\exists$ portal p that is crossed more than twice. Give the tour an orientation. p is crossed at least twice in the same direction.

Cut these crosses and close them on both sides of the portal. This decreases the number of crosses by one.

Correctness $\rightarrow$ exercise $\square$

<u>Claim</u>: No optimal tour has self intersections

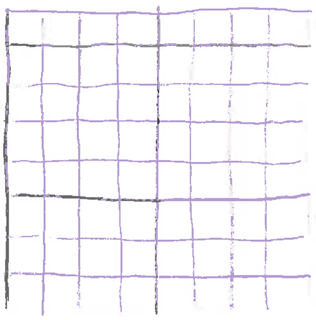
<u>proof</u>: Exercise $\square$

<u>Goal</u>: Show that optimal portal respecting tour is a $(1+\varepsilon)$-approximation.

<u>Problem</u>: Not true

<u>Solution</u>: Shift the lines.

We move vertical lines by an integer a and horizontal lines by an integer b. Such that a vertical line at position x is now at $x+a \bmod L$ and a horizontal line at position y is now at $y+b \bmod L$.



There are L^2 different possible shifts

<u>Lemma</u> Let T be an optimal tour.
There is a shift (a,b) such that there is a portal respecting tour with length at most $(1+2\varepsilon)\,\mathrm{Len}(T)$.

right most and bottom lines are the same as the left most and top lines.

<u>proof</u>. T consists of n straight lines. Therefore, T crosses at most $2\mathrm{len}(T)$ grid lines.

By choosing (a,b) randomly we show that in expectation moving the crossings to portals increases the length by ε per portal crossing. Let a and b be chosen uniformly at random.

The number and locations of the line crossings of T are fixed the level of the lines depends on (a,b).

The number of level i lines is 2^{i} for $i \geq 2$ for level 1 this is 4. Thus for any i the prob of crossing a level i line when crossing some line is at most $2^{i+1}/2L$, since $2L$ is the total number of lines.

The nearest portal of level i at a crossing is at most $\frac{1}{2} L/(2^i m)$ away. Going back and forth through this portal costs at most $L/(2^i m)$.

In expectation the cost of a detour through a portal costs

$$\sum_{i=1}^{k} \frac{L}{2^i m} \frac{2^{i+1}}{2L} = \sum_{i=1}^{k} \frac{1}{m} = \frac{k}{m} \leq \varepsilon .$$

So total cost of the tour T' that is tour T taking detours through portals is in expectation $Len(T) + 2 Len(T) \varepsilon$

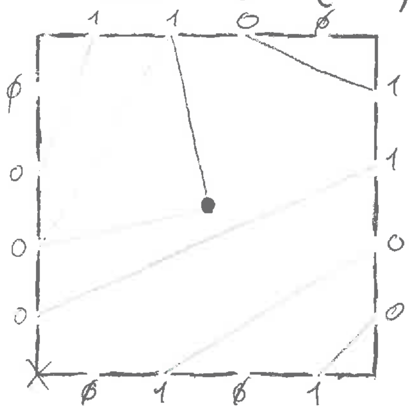
$= (1+2\varepsilon) Len(T)$ $\square$

3. Dynamic program

We have a dynamic programming cell for each combination of:

- shift parameters a, b
- Level i
- Level i square shifted by a, b
- pairs of connected portals.

A level k (1×1) square:



No self intersections of the tour.
Straight lines between pairs.
One pair visits the input point (if there is one).

For any level i square:
At most $4m$ portals (at most m on each side). Each crossed at most twice.
Number of combinations of portal pairs is at most $3^{8m} = n^{O(1/\varepsilon)}$

Why? Imagine there are $8m$ portals (one for each possible crossing). Start at the lower left corner of the square and traverse it clockwise for each portal that is part of a square note 0 if it is first of the pair and 1 if it is last if a portal is not part of a pair note ø. 3 options per portal and this notation gives a <u>unique</u> encoding of each comb. of pairs.

There are n^2 (a,b) parameters
" " k levels
" " 4^i squares at level i $\Big\}$ $O(4^k) = O(n^2/\varepsilon^2)$ total
" " $n^{O(1/\varepsilon)}$ comb. of portals

Total number of cells is $n^{O(1/\varepsilon)}$

Compute the value of a cell:

At level k compute all options to visit the input point (at most $4m$ options).

At level i compute all combinations of the four level $i+1$ squares that it contains.

There are $n^{O(1/\varepsilon)}$ per square: total $(n^{O(1/\varepsilon)})^4 = n^{O(1/\varepsilon)}$

Thus total size of the DP table is $n^{O(1/\varepsilon)}$ and each cell can be computed in $n^{O(1/\varepsilon)}$ for a total running time of $n^{O(1/\varepsilon)}$.